International Journal of Management, IT & Engineering
(ISSN: 2249-0558)

CONTENTS

Chief Patron

Patron

Chief Advisors

Editorial Board

Editorial Advisory Board

Technical Advisors

Associate Editors

Title

ROLE OF ONTOLOGY IN NLP GRAMMAR CONSTRUCTION FOR SEMANTIC BASED SEARCH IMPLEMENTATION IN PRODUCT DATA MANAGEMENT SYSTEMS

Author(s)

Zeeshan Ahmed
Department of Bioinformatics
Biocenter, University of Wuerzburg
Am Hubland 97074, Wuerzburg Germany.

Thomas Dandekar
Department of Bioinformatics
Biocenter, University of Wuerzburg
Am Hubland 97074, Wuerzburg Germany

Saman Majeed
Department of Bioinformatics
Biocenter, University of Wuerzburg
Am Hubland 97074, Wuerzburg Germany

Abstract:

In this research paper we address the importance of Product Data Management (PDM) with respect to its contributions in industry. We also present PDM Systems in brief and highlight some of major challenges to the PDM communities in advance PDM system development. Targeting some of these challenges we present an approach and discuss how this approach can be helpful in solving the PDM challenges. Limiting the scope of this research to one challenge, we focus on the implementation of a semantic based search mechanism in PDM Systems. Going into the details, at first we describe the respective fields i.e. Semantic Web and Language Technology (LT), contributing in ontology construction and natural language grammar implementation, to take advantage in implementing a search engine capable of understanding the semantic output of natural language based search queries. Then we discuss how we can practically take advantage of LT by implementing its concepts in the form of software application with the use of semantic web technology i.e. Ontology. Later, in the end of this research paper, we briefly present the prototype application developed with the use of concepts of LT and Semantic web.

Keywords: Natural Language Processing, Product Data Management, Lexer, Parser, Ontology, Semantic Web

Introduction:

Product data management (PDM) is the computer based system which electronically maintains the organizational technical and managerial data to take advantage in maintaining and improving the quality of products and followed development processes [11, 24]. Major objectives of product data management are to improve the quality of products, improve team coordination, deliver products at the time, reduce engineering environment based problems, provide better and secure access to the configuration based information, prevent error creation and propagation by increasing customization of products, efficiently managing the large volumes of engineering data in reusable form generated by computer based systems and reusing design information.

In 1980s, in the begging of PDM proposition, the concept was new and not very welcomed by the industry of that time but with the passage of time now PDM is becoming more famous and widely in use of many multinational companies. Now a days, most of the PDM based applications are contributing in industry by providing engineering information management module control to access, store, integrate, secure, recover and manage information in data warehouse, distributed networked computer environment based infrastructure, resource management, information structure management, workflow control and system administration [1, 17]. There are many PDM based application developers like Metaphase (SDRC) [7], SherpaWorks (Inso), Enovia (IBM), CMS (WTC), Windchill (PTC) [8], and Smarteam (Smart Solutions) [10].

Like many other communities of different fields the community of PDM System development [18] is also struggling in solving one of currently available challenges i.e. Static and Unintelligent Search. In every PDM System a search mechanism is required and implemented to locate the user's needed information. But unfortunately still there is no such intelligent search mechanism is available which can process user's natural language based queries and can extract the most optimized results in minimum possible time in return [15]. Keeping eyes on discussed problem in PDM System development, we can say, right now PDM community needs a new approach which can be very helpful in implementing the concepts of Natural Language Processing and capable of intelligently handling user's structured and unstructured natural language based requests, process and model the information for fast, optimized and efficient information retrieval mechanism. PDM Systems are mainly designed and implemented for mechanical engineers but as these systems provide complete project, product, process and life cycle management, these can also be used in several difference fields e.g. can be used in auto electrical projects [39], software development life cycle management, product line architecture based applications [42, 43], computational bioinformatics [40] application development and database management [41] etc.

In section 2 of this paper, we present PDM Systems, in section 3 we highlight existing PDM System Challenges, section 4 presents Semantic web and role of ontology in semantic web construction. Section 5 is about Natural Language Processing, section 6 presents a tool for natural language grammar construction, and section 7 presents our proposed approach along with the information about implemented English language grammar in section 8, mapping of grammar

with ontology in section 9 and 10. Section 11 presents the implementation details of proposed approach following prototype discussion in section 12.

PDM Systems:

These systems are developed to manage product data throughout enterprise, ensuring the availability of right information for the right person at the right time and in the right form. PDM systems are mainly used by project managers, designers, engineers, administrators, manufacturing, sales, marketing, purchasing and other personal in the companies [11]. Product-related information controlled by PDM systems includes part definitions and other design data, engineering drawings, project plans, software components of products, product specifications, NC programs, analysis results, correspondence, bills of materials etc. Commercial PDM systems have been developed and are used in companies for more than a decade now. Many companies today have realized strategic importance of a PDM system implementation and usage. But the implementations have often been associated with problems and large costs for the companies. Still there is lot of work to be done in order to improve PDM systems functionality and to develop methods for their proper implementation and use in different areas of the product development and the sales delivery process.

PDM Systems plays an important role in tracking products among different engineering groups by reducing time to market, increasing product quality and reducing total cost. Furthermore PDM System controls, manages and distributes product data automatically to the needed people. A PDM system is typically used within enterprise to organization to access and control data related to its products and to manage the life cycles of those products. PDM Systems are capable of providing user directed and utility functions. User directed functions are i.e. Data vault and document management for storage and retrieval of product information, workflow and process management procedures for handling product data and providing of a mechanism to drive a business with information, Product structure management handling of bills of material, product configurations, and associated versions and design variants, Parts management providing of information on standard components and facilitating re-use of designs. and Program management provides work breakdown structures and allows coordination between product-related processes, resource scheduling and project tracking. Where as the utility functions are the

Communication and notification capabilities such as links to e-mail provide support for information transfer and events notification, Data transport tracking of data locations and moving of the data from one location or application to another, Data translation file exchange in the proper format. Image services, storage, access, viewing and mark-up of product information, System administration system control and monitoring of operation and security.

PDM System Challenges:

Currently the PDM community is facing some major challenges, which are

- Successful implementation of a PDM based application in organization (especially large) because it is time consuming, expensive and most of the staff belongs to corporate management, top level management, engineering management and other engineering and IT professionals do not give importance to PDM and without these person's support it is quite difficult to implement it.
- Time to market benefits of concurrent engineering while maintaining the control of data and distributing it automatically to the respective persons.
- Optimized and decreased cycle time for introducing new technology and products.
- People don't want to use PDM because of several reasons .i.e., they do not want to involve in low level technical and business issues, do not want to spend money, look for fast payback projects, don't have time, too much inertia in this company, lack of trust of users on management, job insecurity, incapable of handling PDM systems, CAD, PDM is immature, not flexible, its risky, not intelligent to get the right information, limited in memory, electronic and dependent on system, slow.
- PDM Systems are required to be platform independent because in the new business model, it is nearly impossible to mandate that all the potential users choose the same platform or the same operation system.
- PDM Systems are required to be easily extendable because whenever new features are demanded users must reinstall or upgrade the client application completely.

- Traditional PDM Systems are not adequately available, secure, reliable, and scalable for global enterprise services.
- Standardized Web based frameworks for Web based PDM systems development.
- PDM System cannot provide access to users at different locations, especially those on different networks. In each implementation of the PDM, the network configurations must be homogeneous. Hence, a bottleneck emerges for global companies wishing to implement PDM [27].
- The PDM client applications are platform dependent, which means either all users must use the same computer platform or a specific application must be provided for each user platform [27].
- In the new business model, it is nearly impossible to mandate that all the potential users choose the same platform or the same operation system [27].
- The current PDM system is not easy to extend. Whenever new features are demanded, users must reinstall or upgrade the client application completely. Therefore, it increases the maintenance cost when a company is becoming globally based [27].

Semantic Web:

Targeting the challenge of implementing a web based product data management system capable of performing semantic based search to extract desired information from attached repositories over the web, the field of Web and Semantic Web is explored, as it promotes the implementation of semantic based web applications by providing the concept of structuring of data over the web to take advantage in extracting semantic based information. World Wide Web is a global information sharing and communication system made up of three standards: Uniform Resource Identifier (URL), Hypertext Transfer Protocol (HTTP) and Hypertext Mark-up Language (HTML) by Tim Berners-Lee to effectively store, communicate and share different forms of information. The Information is provided over the web in text, image, audio and video formats using HTML, considered unconventional in defining and formalizing the meaning of the context.

Semantic Web is a mechanism of presenting information over the web in a format so that human being as well as machines can understand the semantic of context. Semantic web is a mesh of information which can be linked up in a way, so that it can easily be processed by machines [27] and aim to produce technologies capable of reasoning on semi structured information [29]. The semantic web is an intelligent incarnation and advancement in World Wide Web to collect, manipulate and annotate information independently by providing effective access to the information. Semantic web provides categorization and uniform access to resources, promoting the transformation of World Wide Web into semantically modelled knowledge representation systems and common framework which allows data to be shared and reused [33]. Semantic web also gives the concept of semantic based web services to provide solutions to the problems of dynamically composed service based applications.

Currently, semantic web is standing on a very important building block: Ontology [28]. Ontology is a main building block of Semantic Web to provide the information in machine processable semantic models and produce semantically modeled knowledge representation systems. It is playing a vital role in solving the existing web problems by producing semantic aware solutions. Ontology makes machines capable of understanding the semantic of languages that humans use and understand by producing the abstract modeled representation of already defined finite sets of terms and concepts involved in intelligent information integration and knowledge management [34]. Ontology is basically categorized in three different categories i.e., Natural Language Ontology (NLO), Domain Ontology (DO) and Ontology Instance (OI) to provide relationships between generated lexical tokens of statements based on natural language, knowledge of a particular domain and to generate automatic object based web pages [30]. Ontologies are constructed and connected to each other in a decentralized manner to clearly express semantic contents and arrange semantic boundaries to find out required needed information [35].

Natural language based information is treated as the input to the ontology construction process, which parses the text in nouns and verbs. Nouns are represented as “Classes” and verbs as “Properties” containing values, relationships with other properties and some constraints. Classes are further divided in main and sub class categories maintained in taxonomical hierarchy. The size of ontology varies due to the increase in number of classes and instances. Ontologies can be made manually from scratch, by extracting information from web and by merging already

existing ontologies into new ontologies. But this manual process sometimes becomes very complex and time consuming especially when dealing with the large amount of data. Moreover, to support the process of semantic enrichment reengineering for the building of web consisting of meta data depends on the proliferation of ontologies and relational meta data. This requires high production of meta data at high speed and low cost. So in these cases machine learning approaches can be very helpful in generating ontologies automatically because they provide real time schemes like classification rules, instance based learning, numeric predictions, clustering, Bayesian networks and decision trees which can be very helpful in the generation of ontologies.

First step in building ontologies is to create the nodes and edges. Once the concepts and relationships of graph based ontology are constructed then next step is to quantify the strengths of semantic relationships [31]. Ontologies can be constructed manually and automatically by using some ontology supporting languages i.e., XML (eXtensible Mark-up Language), RDF (Resource Description Framework) [27] and OWL (Web Ontology Language) offering ways of more explicitly structuring and richly annotating Web pages.

The development of ontology driven applications is difficult because of some limitations and principal problems which are as follows

- Natural language parsers used to parse the information to construct the ontologies are limited because they can only work over a single statement at a time [8].
- Existing methodologies of structuring ontologies are insufficient and need to be improved because now it is quite impossible to define the boundaries of ontology based particular domain's abstract model and automatically handle the increase in size of ontology due to the increase in number of classes and instances.
- Creating ontologies manually is a time consuming process which becomes very complex when there is a large amount of data to create large number of ontologies from. To take advantage in creating large number of ontologies by reducing the complexity and time, an automatic ontology creation mechanism is required. Some mechanisms are already proposed and implemented to create ontologies automatically but they are insufficient and less qualitative. While creating nouns based classes using existing automatic ontology creation mechanism, it is quite impossible to identify the possible existing relationships between classes to draw the taxonomical hierarchy [36]. Furthermore it is

also quite impossible to perform automatic emergence of ontologies to create new ontologies [38].

- Currently available ontology validators are restricted and not capable of validating all kind of ontologies e.g. based on complex inheritance relationship [32].
- Domain specific ontologies are highly dependent on the domain of the application and because of this dependency domain specific ontologies contain specific senses which are not possible to find in general purpose ontology [37].
- The process of semantic enrichment reengineering for web development consists of relational meta data required to be developed at high speed and in low cost depending on proliferation of ontologies, which is currently also not possible.
- Handling the dynamically raised calculations caused by the comparison of big complexities of similar ontologies is also not possible [38].

Natural Language Processing:

Today one of the most targeted problems in the field of artificial intelligence (computer science) is to make machine this much intelligent so then it can almost behave like a human being. Some of the behaviors of human beings have been accomplished during machine implementation e.g. now days machines can hear with the use of microphone, speak by producing sound, see with the use of cameras, smell with sensors but still there are some areas where this machine development is not completely successful and some of them are to understand natural language, learning from experience and making autonomous decisions in real time environment etc. In this research paper we will not discuss the moral problems which arise when it comes to machine development with intellectual capacities that rivals human beings but we will handle with the problems of how to make the machine understand the human languages. Natural Language Processing is one of the major step towards the development of field Artificial Intelligence as it deals with the propositions consisting of the production ability to implement an intelligent system which can not only process information but also can understand the user instructions in natural language e.g. English. Natural Language Processing performs two major tasks i.e. Natural language

understanding (NLU) and Natural language generation (NLG) as shown in Figure 1, to full fill the set goals of human machine communication implementation [44, 45].

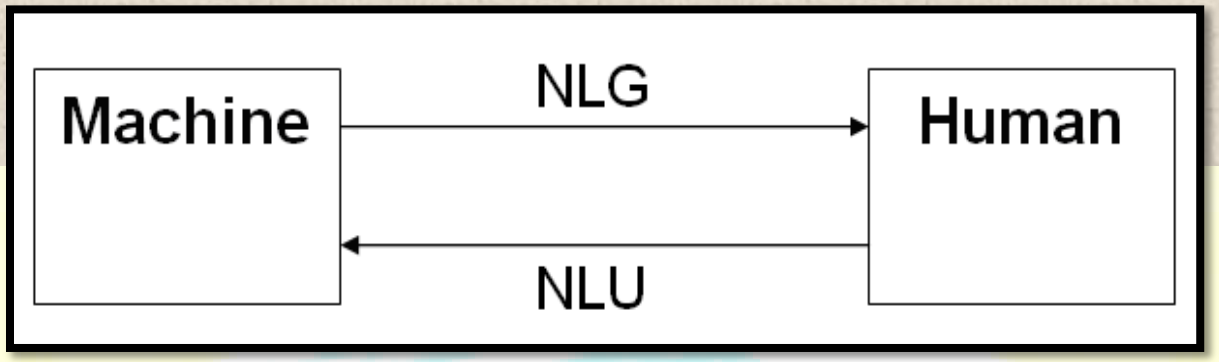

Figure.1. Human Machine Communications [44, 45]

These tasks are implemented in two ways i.e. processes of converting the information from natural language to machine language and then vice versa converting the machine readable data to natural language based instructions. The main problem regarding this communication is the structuring and restructuring of information with respect to the grammar of natural language and programming language involved. Deep down at basic (low) level computers can only understand binary instructions to process with but here as we are dealing with the problem to make machine understand natural language, we need a natural language processor to translate languages from natural language to binary language and then back from binary to natural. To take advantage in the implementation of our targeted solution, we have used some programming language's concepts, whose compilers with the inclusion of assemblers are already doing the job of converting the information to binary level and then returning back the information after converting back to natural language statement (strings) from binary. To meet aforementioned goals, compiler consists of two main components i.e. Lexer and Parser. Lexer is basically a lexical analyzer, based on the dictionary of the defined tokens of concerned language. Where as the Parser is the semantic analyzer based on the rules defined in the grammar.

NLP is one of the subfield of artificial intelligence, claiming the jobs of mainly analyzing, understanding and generating human (natural) languages, following the three step language conversion procedure as shown in Figure 2, [44, 45]. Likewise a programming language's compiler/interpreter, NLP consists of mainly three components i.e. Lexical Analyzer,

Semantic Analyzer, and Translator. The lexical analyzer is also known as lexer, defining symbols or separate groups of symbols from the phrase [5]. It works like a filter program, which searches for certain characters by breaking actual character based statement(s) into tokens e.g. symbols, letters, digits, constants, reserved words, whitespace and comments etc., to perform certain defined tasks [4]. Furthermore it also regroups the input as series of characters with group significance i.e. tokens. Token is a symbol or group of symbols, don't have much sense at this level but they acquire meaning in the next step namely parser. The meaning of the tokens is given from the lexer rules. The stream of tokens generated by the lexer is received by the Semantic Analyzer to further process with. The Semantic Analyzer is known as a Parser which receives the input source program and breaks these instructions into parts. It groups the tokens received from the lexer according to the given language grammar. If the grammar doesn't contains this group of tokens, the parser denies the successful processing otherwise proceed with.

There are two types of parser i.e. LL and LR parsers. The difference between them is in the derivations. The LL parsers construct a leftmost derivation of the input and LR constructs the rightmost derivation. This means that the LL parser replaces the left-most non terminal first and LR replaces the right most non terminal first, but both of them parses from left to right. Furthermore parser creates the sequences of tokens to put them into an Abstract Syntax Tree (AST) and makes one or many tables with information about the tokens or group of tokens i.e. symbol table, used to validate the types of the data. Residing with in the domain of Natural Language Processing, several approaches have been introduced by many researchers which are providing lots of values in the implementation of natural language processing based application by writing Grammar and with the implementation of lexer and parser including Analyzing English Grammar [19], Layerd Domain Class [20] and Another tool for language recognition [3].

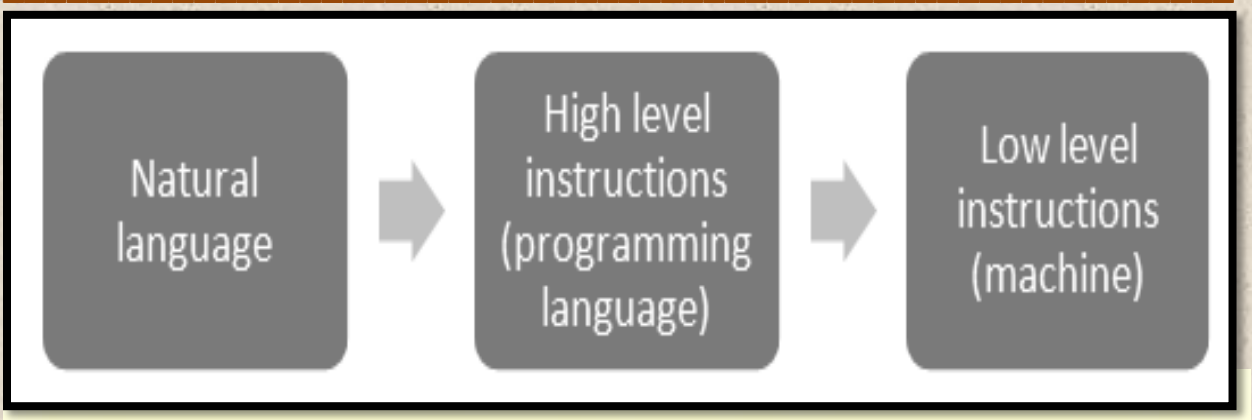

Figure.2. Three Step Language Conversion Process [44, 45]

ANTLR:

ANother Tool for Language Recognition (ANTLR) is a tool, developed in 1983 [3] by Professor Terence Parr and his colleagues to write grammar for both lexer and parser. It is implemented in Java but it can generate source code in Java, C, C++, C#, Objective C, Python and Ruby. ANTLR use EBNF (Extended Backus-Naur Form) for the grammars, which is very formal way to describe the grammar. ANTLR provides a standard editor for grammar writing and generating lexer and parser. Till now this tool has been used for programming language's grammar writing but we are considering it for natural language processing by writing natural language's grammar and generating lexer and parser to make the machine understand it.

ANTLR has many belonging applications and opportunities to extensibilities, as shown in Figure 3. One of the biggest benefits is the grammar syntax; it is in EBNF form, which is a Meta syntax notation. Each EBNF rule has a left-hand side (LHS) which gives the name of the rule and a right-hand side (RHS) which gives the exact definition of the rule. Between the LHS and RHS there is the symbol ":" (colon), which separates the left from the right side and means "is defined as". Another benefit is the graphical grammar editor and debugger called ANTLRWorks, written by Jean Bovet and gives us the possibility to edit, visualize, interpret and debug any ANTLR grammar.

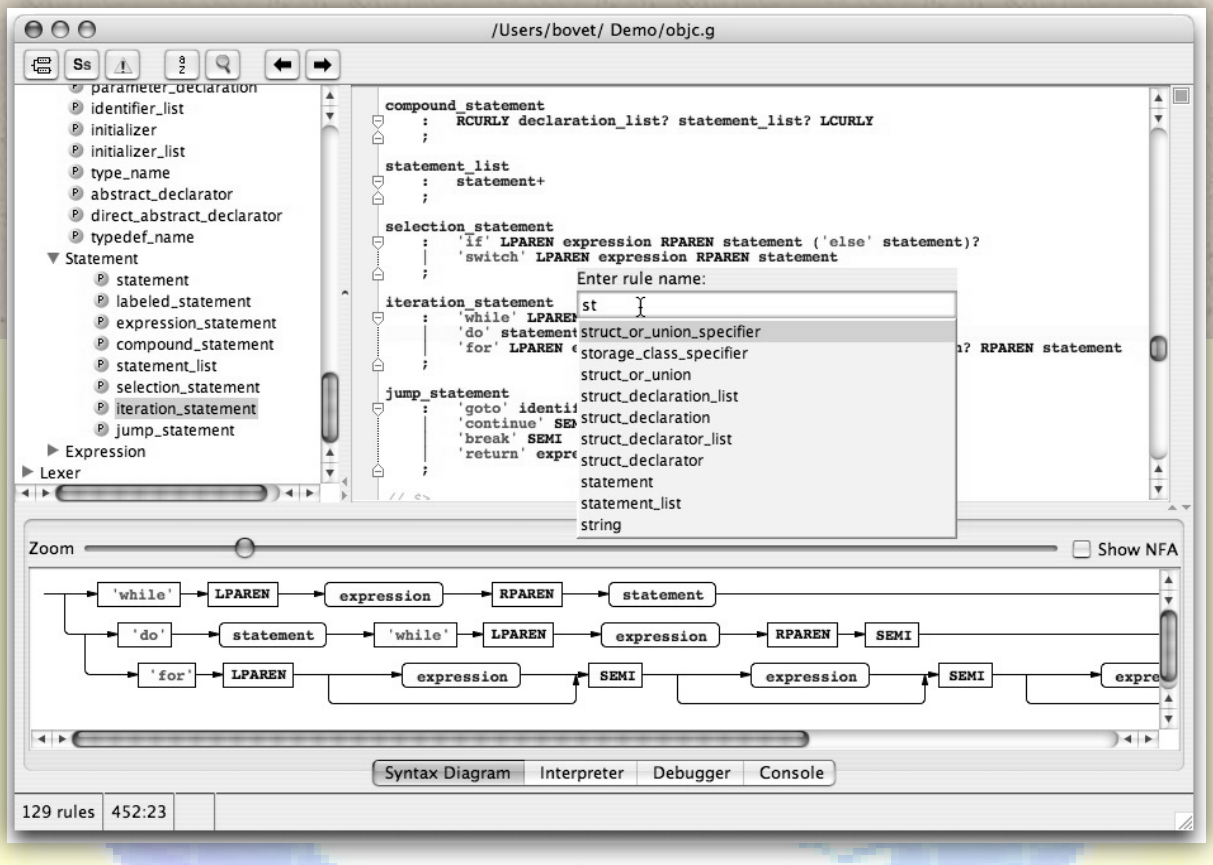

Figure.3. ANTLR Screen Shot [21]

Approach:

To take advantage in solving the problem of implementing of an intelligent human machine interface consisting of intelligent user system communication, meta data extraction out of unstructured data, semantic oriented information modelling, fast managed data extraction and final user end data representation an approach Intelligent Semantic Oriented Agent Based Search (I-SOAS) [1, 12, 25] has been proposed. The proposed conceptual architecture of I-SOAS is consists of four main sequential iterative components i.e. Graphical User Interface (GUI) [2, 16], Search [5, 6, 15], Data Management (DM) [26] and Data Representation (DR), as shown in Figure 4.

Figure. 4. Intelligent Semantic Oriented Agent based Search Conceptual Architecture [12]

GUI is responsible for the intelligent user system communication. GUI is proposed as a flexible graphical user interface capable of first analyzing the source of input, forwarding inputted data for further processing and responding back to the user with end results. Moreover GUI is supposed to be flexible enough so then it can be learned in shortest possible time and redesigned by user itself according to his need and wish. To implement the GUI as shown in Figure.4, GUI is divided into two main sub-categories i.e. Graphical User Interface and Communication Sources. Graphical User Interface consists of the concept of three more sub-

categories i.e. Intelligent, Flexible and Agent to intelligently handle the user's unstructured requests, provide multiple options to redesign the graphical user interface according to the ease of the user by user itself and perform internal architectural component's agent based communication. Where as in Communication Sources, first the corresponding user is supposed to be identified to enable the correct communication mode, if it is a digital system then electronic data communication mode will be enabled and if it is natural system then natural language based communication mode will be enabled.

Search unit is the most important component of the I-SOAS. The quality of performance of I-SOAS depends upon the accuracy in the results produced by this component. The overall job of Search is divided into five main iterative sequential steps i.e. Data reading, Tokenization, Parsing, Semantic Modelling, Semantic based query generation. The main concept behind the organization of these five steps is to first understand the semantic hidden in the context of natural language based set of instructions and generate a semantic information process able model for the system's own understanding and information processing. In the first step, Data Reader is supposed to read and organized inputted data from GUI into initial prioritized instructions list. Then in the second step Data Tokenizer is supposed to tokenize instruction one by one, which are then treated in the third step by Data Parser for parsing and semantic evaluation with respect to the grammar of used natural language. Then in the fourth step Semantic Modeler is supposed to first filter the irrelevant semantic less data and then generate Meta data based semantic model. Then in the last and fifth step Semantic Based Query Generator is supposed to generate a new query used for further data storage and extraction of desired result.

DM is responsible for two main functions i.e. Semantic based Query Processing and Database Management. Semantic based query built in Search is treated by Semantic based Query Processor to generate SQL query to run in to database to store and extract the required information. The job of Data Manager is to manage the processes of SQL query building, data extraction and creation of new indexes and storage based on newly retrieved information. DR is responsible for responding back to the user with finalized end results. This component consists of six sub components i.e. Information Retriever, Information Reader, Information Tokenizer, Information Parser, Information Reconstructor and Presenter. The job of this component is somehow similar to the job of SEARCH, but major difference is of handling data and information. Search treats data to process but DR treats information. Required extracted and

managed information from DM is passed to DR using Information Retriever, which simply read and organized by Information Reader without performing any analytical action except the prioritization of informative statements. Then using Information Tokenizer and Information Parser statements are tokenized and parsed, then using Information Reconstructor finalized formatted information is supposed to be built in user's used natural language based grammar. Finally Presenter presents the resultant information to GUI to respond back to the user.

I-SOAS NLP Grammar:

We have written and designed the grammatical view for the I-SOAS Lexer and I-SOAS Parser. This grammar is based only on English (Natural Language). At the moment, according to the scope of our research, the proposed grammar has been divided into three main categories i.e. A, B and C [44, 45], as shown in Figure 5.

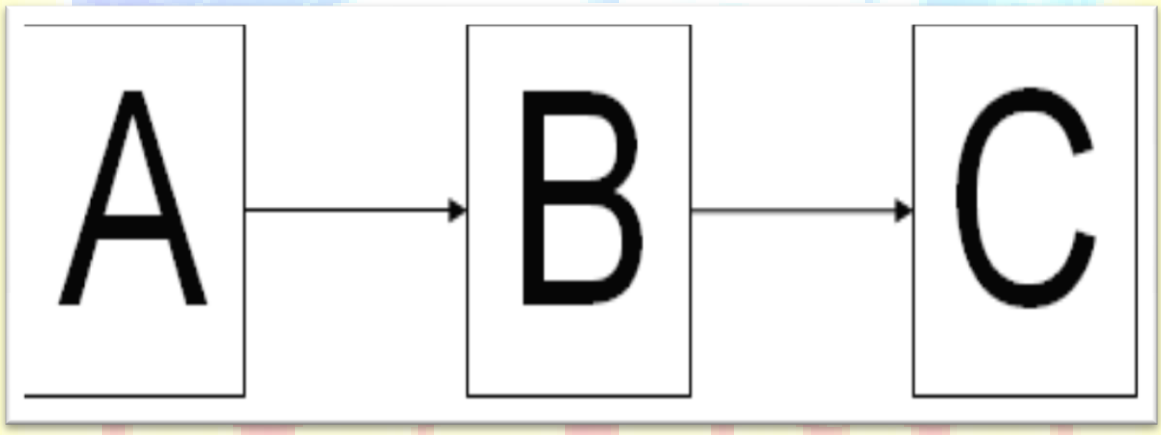

Figure.5. I-SOAS Lexer and Parser Grammar [44, 45]

A category is representing all English words belonging to the User (requesting for some results), B category is representing the English grammar structure and C is representing the main Object user is looking for. This grammar consists of 15 different lexer rules and 11 different parser rules in total.

Mapping of I-SOAS Grammar to Ontology:

One of the steps from Information Processing part in I-SOAS is the Semantic Modeler. I-SOAS Modeler makes a semantic model of the grammar and saves it into the Database (this is the connection between Search and DM). After the information is forwarded to the Resolver, which puts the relationships between the statements in the semantic model and also saves this into the Database. To make a conceptual model of the grammar we should use the ontology theory, which will describe its structure. Ontology clarifies the semantics of a conceptual modeling grammar [7]. In computer science the word ontology means to show one area of knowledge with conceptual scheme. The most used definition of ontology is from Gruber

"Ontology is a formal, explicit specification of a shared conceptualization" Gruber (1993)

Formal means that the specification should be machine processable, explicit means that the elements should be simply defined and specification is an abstract model. According to Gruber the ontology is such a representation of a domain, where a set of objects and their relationships is described by a vocabulary [9]. There are different kinds of ontologies but we will use Natural Language Ontology (NLO) to provide the relationships between the different statements in our grammar [8]. There is also Domain Ontology (based on the knowledge of a particular domain) and Ontology Instance (which generates automatic object based web pages), all are connected to extract hidden semantic out of data.

This Ontology scheme is made from fields with the data and connections between the different objects, which declare the rules in the knowledge area. We can make an ontology scheme for our grammar to show how rules can be constructed by connecting the words. It includes a dictionary and logical relations between the terms and their meanings, the treatment of one term with others, different variations between them and a semantic model of the terms. The ontology construction process begins with the parsed text from the natural language i.e. Words. The nouns in the ontology are called classes and the verbs are the properties, having different values for the different relationships. The first thing to do is making an oriented graph with all the classes and properties with the semantic relationships in between, then the ontology can be made with some of the ontology supporting languages like XML (eXtensible Mark-up Language), RDF (Resource Description Framework) and OWL (Web Ontology Language) [8] etc.

Residing in the domain of Semantic Web many products are available and several approaches have been introduced by many researchers which are providing lots of values in the implementation of semantic based applications with use of Ontology and providing structured data over the web to take advantage in implementing efficient web based information retrieval search mechanism e.g. Semantic Dektop Personal Information Model (PIM) [22], Meta Data Search Layer [23] etc. In this section, without going in to much product detail, we will only present some semantic web based approach, to take advantage in having an idea about semantic based system development using Ontology.

The map of our ontology should explain the relationships between the words in the grammar, what word comes after another. To answer the question "why", we will explain the English grammar. There are two types of verb tense that are used in our grammar i.e. Present Simple and Present Continuous. With them we have also two different persons and two different figures i.e. singular and plural. This explains the statements from A to I i.e. here we are answering the questions: "Who", "What is he doing" or "What does he do" etc. Of course in the real search we can not use the personal pronoun because we can just say the action. That is why we have statements A D K, B D K, C E K, but also statements without A, B and C (which are the pronouns). D K, E, K is the example for Present Simple Tense, but we have it also for Present Continuous Tens. We have also statement J, which never uses a pronoun with itself (* J K) [44, 45].

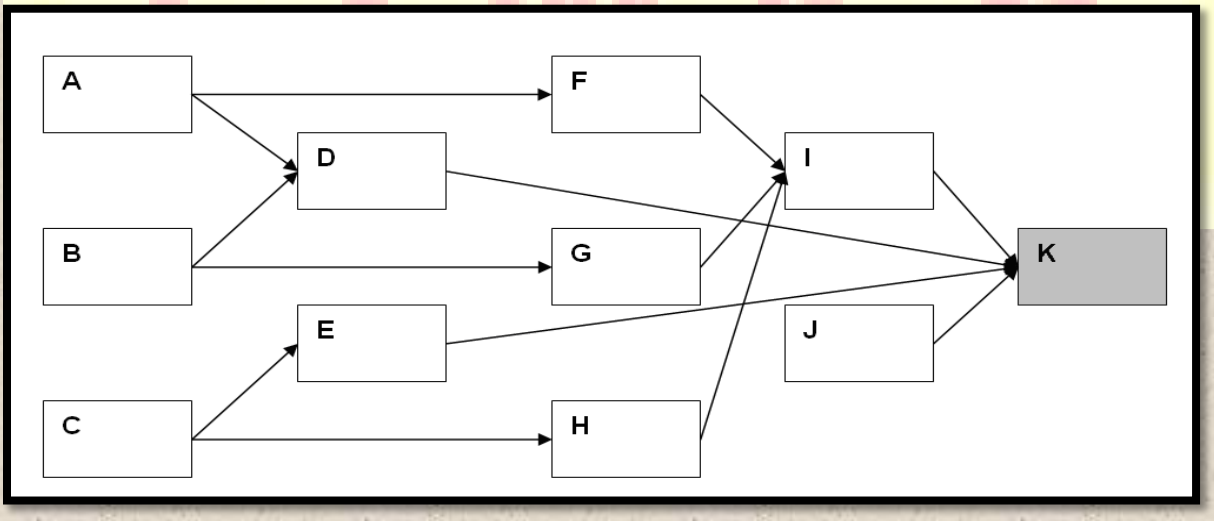

Figure.6. NLP Grammar Ontology Mapping [44, 45]

In the picture of our ontology you can see the exact connections between the statements. There are in way just form left to right and there is no need to start always from A, we can start any sentence from where we want but follow the directions. The only mandatory statement is K, a key word, which defines what we want to search. In the following map scheme you can see the grammar with the connections between the words. As shown in Figure 6, to produce ontology based search we have mapped the used grammatical view in Lexer and Parser in to an ontological view. There three main entities i.e. A, B and C. There are five main properties of A i.e. I, We, He, She and They, six properties of B i.e. am, are, is, looking/searching for, need(s)/want(s) and unknown and C has only one property K.

I-SOAS Ontology:

Following the concepts of semantic web and ontology construction we have created the following ontology with respect to the grammar of I-SOAS, as shown in Figure 6. Created I-SOAS Ontology As shown in Figure 7 consists of one main Class ISOAS, then three sub classes of ISOAS .i.e., are A, B and C. All three subclasses contain their further subclasses and the relationships of this subclass with each other. Class A contains five sub classes .i.e., I, We, He, She, You, It, This, That and They. Class B contains seven sub classes .i.e., is, are, am, need, want, where and D, D is another sub class containing two more sub classes .i.e., looking and Searching, and looking has another sub class for. Where also contain two more subclasses .i.e., between and equal. Class C contains only one sub class K.

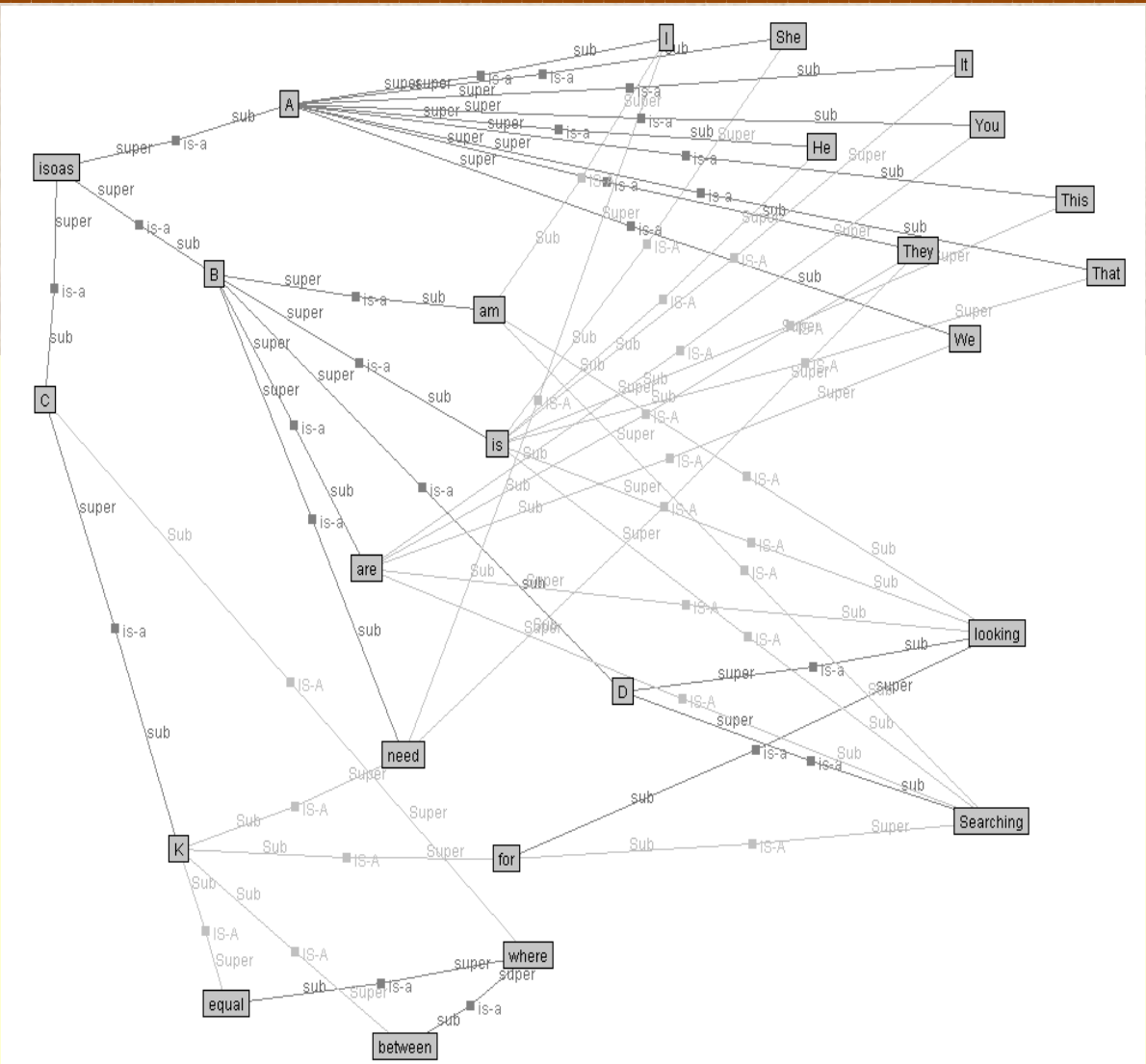

Figure.7. I-SOAS Ontology; Class relationships [13], [14]

Class A has direct relationship with its subclasses

- relationship with "are"
- "We" has the relationship with "are

Class B has direct relationship with its subclasses,

- "is" has the relationship with "looking" and "searching"
- "am" has the relationship with "looking" and "searching"

- "are" has the relationship with "looking" and "searching"
- "need" has the relationship with "K"
- "need" has the relationship with "Where"
- "want" has the relationship with "Where"
- "want" has the relationship with "K"
- "where" has the relationship with "between"
- "where" has the relationship with "equal"

Class C has direct relationship with its subclasses "K", and "K" has relationship with "need", "want" and "for".

I-SOAS Implementation:

NLP Search Module is designed to read, organize, tokenize, parse, semantically evaluate, model and process data. The theme behind this module is to develop a system which can process user's natural language based requests to extract needed information from attached repository and return obtained results back to the user. NLP Search module is designed to provide an intelligent search capable of processing natural language based queries and providing query builder to write and run SQL queries to extract stored results. As this is a prototype application, the reason of providing SQL query builder is to enable technical user for writing direct SQL queries to validate the obtained results using natural language based search.

To aforementioned design requirements, a flow char is designed, presenting the internal work flow of NLP Search. The flow chart of the NLP Search module, as shown in Figure 8, starts with input text which is first stored in the database and then forwarded to the Lexer. The Lexer is to first tokenize the whole input text into possible number of tokens, and then evaluate the context with respect to the used grammar of the input language. These token are then forwarded to the Parser, which considers all the statements or instructions as the combination of all tokens produced by the Lexer for semantic evaluation. Resultant semantically evaluated information from Parser is forwarded to Modeler. Modeler produces semantic models using

syntactically and semantically evaluated information. Then generate SQL queries to extract data from attached database Then finally outputted information is presented to the user.

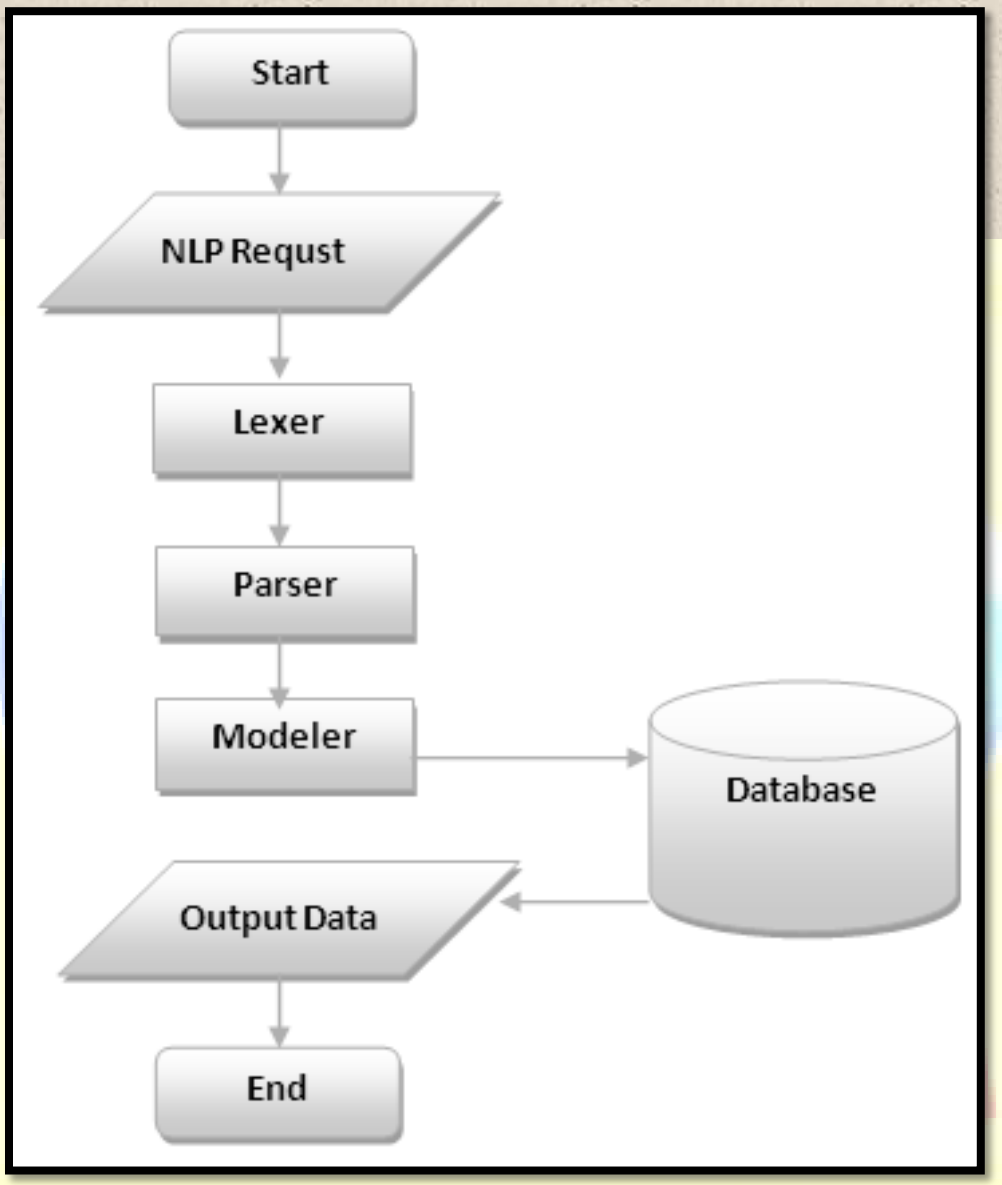

Figure .8. I-SOAS Flow Chart of NLP Search

The use case diagram shown in Figure 9 provides the description of a system behavior of developed Natural Language Search. This use case diagram presents the number of events a user can trigger using developed GUI and system responses against those fired events. Moreover it

also provides the information about prerequisites, expected and actual behaviors and basic flow of the system, presented in Table 1.

Table. 1. I-SOAS - Use Case NLP Search

Use Case Elements	Descriptions
Use Case Number	1
Application	NLP Search Prototype
Use Case Name	NLP Search Module
Use Case Description	This use case describes the NLP Search module. Going into the details it describes the number of possible actions can be performed by user using this application and expected results in return by the system.
Primary Actor	User
Preconditions	1. Application should successfully be running on server 2. Adobe Flash should be installed in the system 3. Application should be browsed using any web browser 4. User must be log in by entering valid user name and password
Trigger	1. This use case triggers the following events 2.. Take Natural Language based User Input 3. Take SQL based User Input 4. Analyze User Input 5. Extracts results from Database 6. Presents output 7. Logout or Quit
Basic Flow	1. The basic flow consists of following user steps of actions 2. User must login. 3. Enter natural language or SQL based request to search desired

	results. 4. In return extracted results from connected repository will be presented to the user. 5. User can logout or quit the applications
Alternative Flow	There is no alternative flow or exception for this User case
Expected Results	1. This use case triggers the following events 2. Take Natural Language based User Input 3. Take SQL based User Input 4. Analyze User Input 5. Extracts results from Database 6. Presents output 7. Logout or Quit
Actual Results	1. This use case triggers the following events 2. Take Natural Language based User Input 3. Take SQL based User Input 4. Analyze User Input 5. Extracts results from Database 6. Presents output 7. Logout or Quit
Expected Result	Successful usage of NLP Search
Actual Results	Successful usage of NLP Search
Found Exception	No

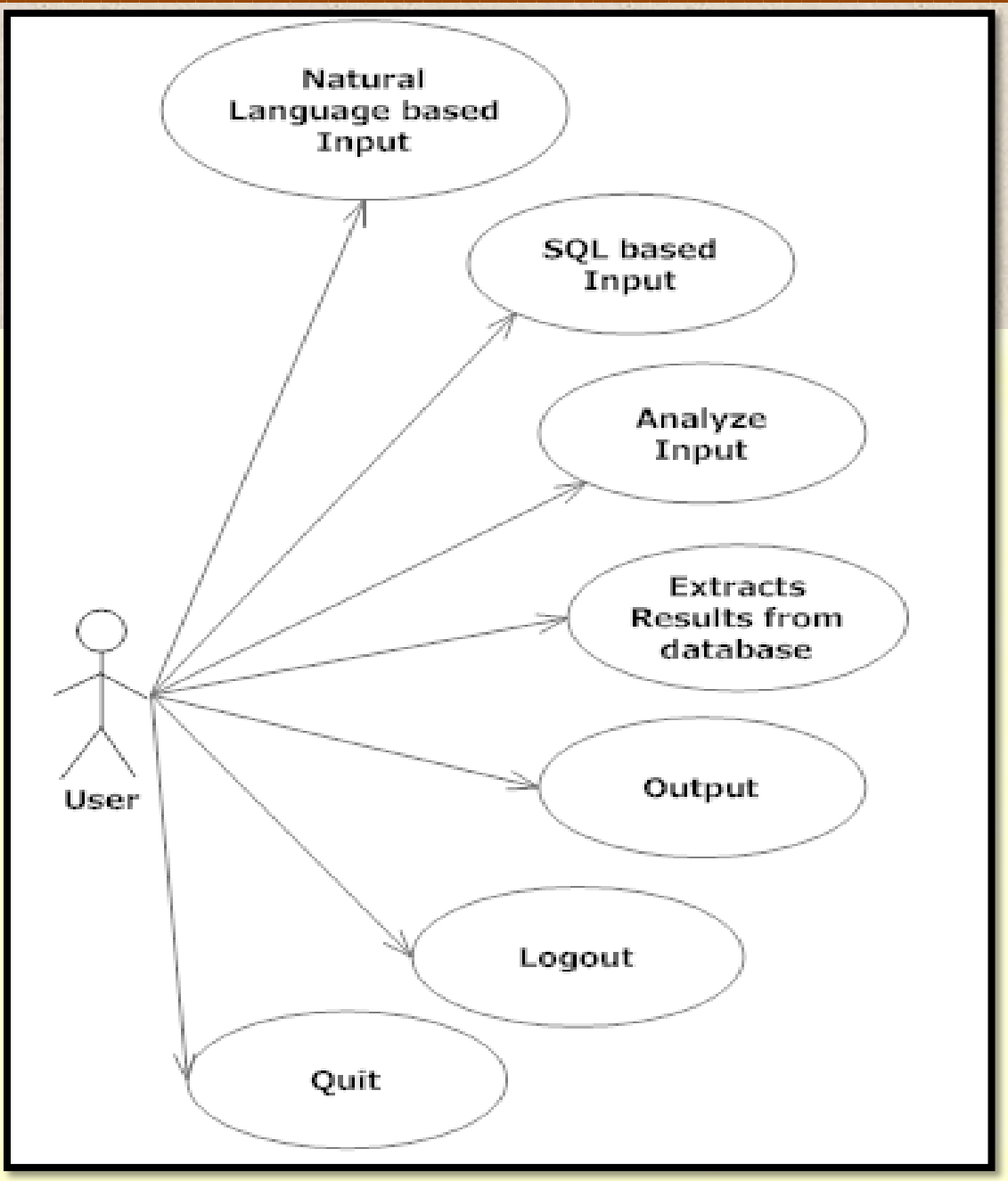

Figure. 9. NLP Search Use Case

I-SOAS Prototypes:

Here, in this section, we present I-SOAS Tool for information engineering & modeling, warehousing and knowledge base. Narrating the abilities of currently available demo version we describe the major capabilities of the available I-SOAS demo version. Following the constructed

implementation designs and meeting the design requirements, the following prototype version of I-SOAS has been developed, as shown in Figure 10.

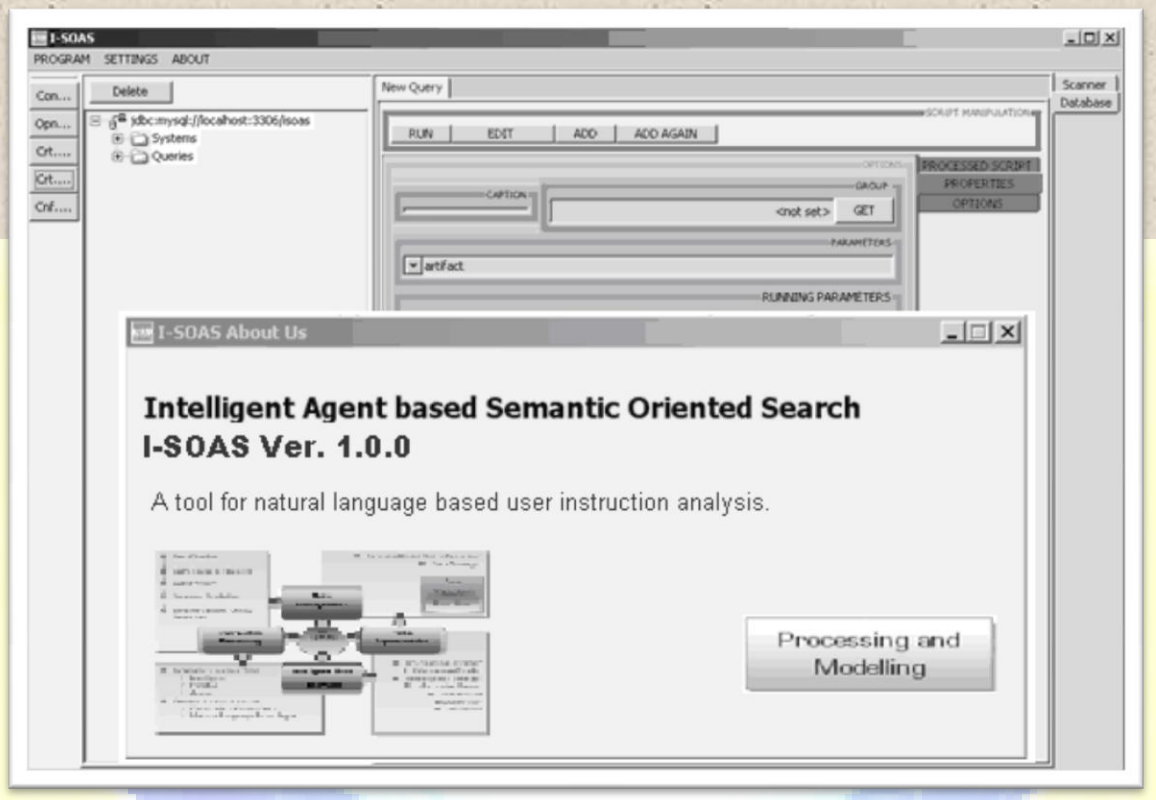

Figure.10. I-SOAS Desktop Prototype Version 1.0.0

The currently available version of I-SOAS Desktop Application is capable of

- Running as a stable application
- Taking input from user in the form of text file
- Providing flexibility in the organization and presentation of available options in the graphical machine interface
- Creating dynamic database to store user input based information
- Connecting and disconnecting with different databases
- Opening database to view the stored information
- Configuring files based on their types (extensions)
- Applying Lexer to input (file(s))

- Applying Parser to input (file(s))
- Storing resultant information in database
- Building queries to extract stored results
- Producing different kind of visualizations of stored user based and system processed information in database

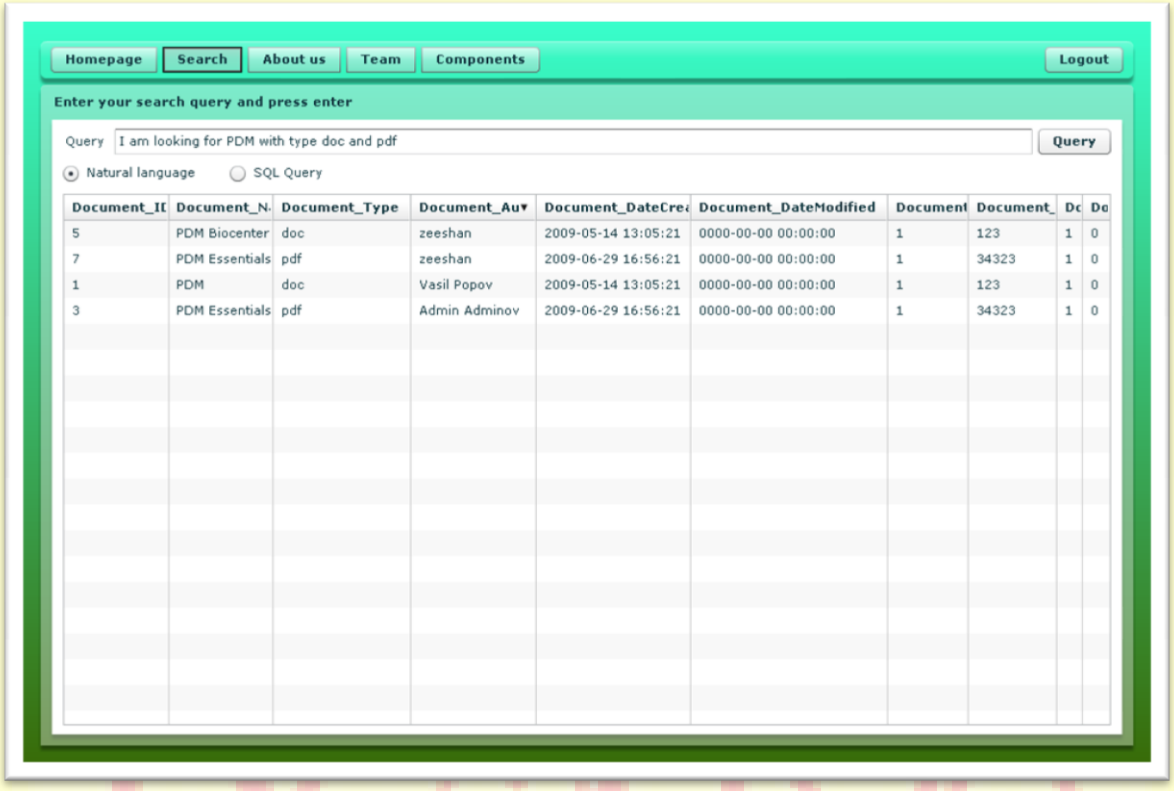

Figure.11. I-SOAS Web Prototype (1) Version 1.0.0

To meet aforementioned jobs, I-SOAS is divided into ten different front end screens .i.e., Information Processor, Create Database, Open Database, Connect Database, File Configuration, Relation Builder, Query Builder and Result Visualizer. Information processor is capable of mapping directory information into tree, opening database interface of I-SOAS, Inputting process able file types; Exporting SQL based information and providing views of database. Create Database allows user to create new repository to store, manage and manipulate processed

and preprocessed user inputted data. Open Database allows user to select database (amongst available database).

Connect Database establishes database connections. File Configuration is the graphical machine interface for to selecting file type(s) to select and process by information processor, furthermore it also allows user to select parsers, in case of more than one language processors (at the moment this option is limited to only one language i.e. English). Relation Builder is the interface to add, edit, run and save query script. Query Builder is the graphical machine interface to set the caption of the query, get query properties, select already stored query to use and run the query to extract information from database. Result Visualizer Allows user to select visual type (charts, graphs, maps etc.) and style (vertical, horizontal etc.) and visualizes resultant information in decided visualization mode.

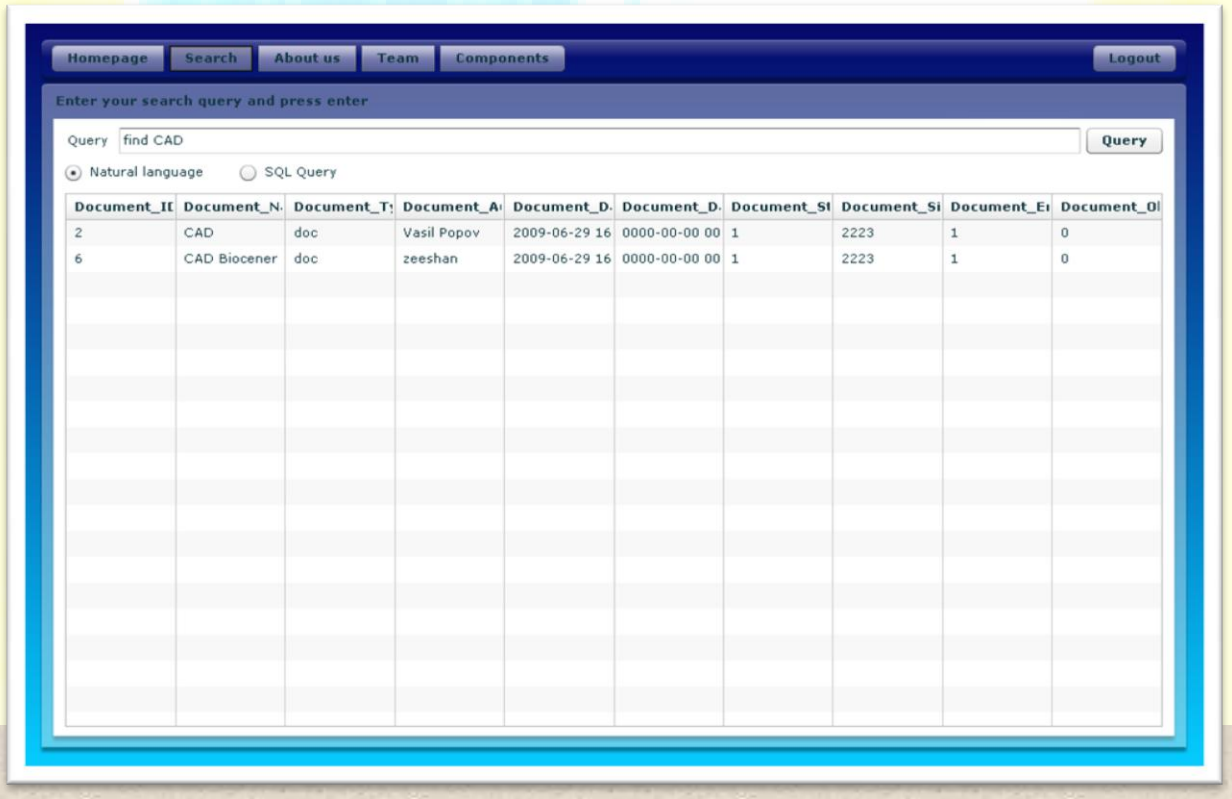

Figure.12. I-SOAS Web Prototype (2) Version 1.0.0

Later the core developed portion (language processor) of this desktop version of I-SOAS is incorporated in the recently developed web release of I-SOAS [16], as shown in Figure 11. Web system is processing following natural language based query i.e. I am looking for PDM

with type doc and pdf, and the resultant information is presented in Figure 11. Furthermore, one another natural language (two keywords) based instruction is processed i.e. fine CAD, and the resultant information is presented in Figure 12.

Conclusions:

In this research paper we have briefly addressed the importance of Product Data Management (PDM) along with currently available major challenge of natural language based search towards the Product Data Management System Development. Targeting the challenge, in this research paper, we have discussed a way for the implementation of a semantic based search by writing a natural language based grammar with the use of ANTLR and later mapped that grammar into ontology. As this is an ongoing research, we are hoping for a web based product data management system development in future, with the use of approach discussed in this research paper. We have offered a way for the machine to understand natural languages and also promote the artificial intelligence. This could be a step for the newest communication technologies. We have also compared our proposed approach with some existing approaches and concluded with some more beneficial results, in both ways, by comparing languages processing search mechanism [15] and graphical user interface [16].

Future Recommendations:

A new grammar for natural language processor implementation for PDM Systems have been proposed and written but due to the limited scope, so far, of this research and development work, the word length of proposed dictionary for lexer is small and the number of rules designed for parser are also a few. In future, looking forward to extend the scope of this and add more rules and tokens to the dictionary of the grammar and remap using ontology. Further research is in process to develop a semantic based efficient natural language processor to execute complex queries.

Acknowledgement:

We (authors) are thankful to the University of Wuerzburg Germany for giving us the opportunity to keep working on this research project. We are thankful to the blind reviewers and publishers for publishing this manuscript.

References:

- Ahmed. Z, D. Gerhard: "Semantic Oriented Agent based Approach towards Engineering Data Management, Web Information Retrieval and User System Communication Problems", In the proceedings of 3rd International Conference for Internet Technology and Secured Transactions, ICITST 08, IEEE - CST, June 23–28 2008
- Ahmed. Z. 2009. Intelligent human machine interface design for advanced product life cycle management systems. In Proceedings of the 6th international Conference on Frontiers of information Technology (Abbottabad, Pakistan, December 16 - 18, 2009). FIT '09. ACM, New York, NY, 1-4. DOI= http://doi.acm.org/10.1145/1838002.1838058
- Mark Volkmann, "ANTLR 3", Object Computing, Inc., 2008
- Programming language,last reviewed 22 June 2010, http://en.wikipedia.org/wiki/Programming_language
- ANTLR, Ashley J.S Mills, Last reviewed 27.06.2009, < http://supportweb.cs.bham.ac.uk/docs/tutorials/docsystem/build/tutorials/antlr/antlr.html>
- Ahmed. Z, S. Majeed, T. Dandekar: "Towards Design and Implementation of a Language Technology based Information Processor for PDM Systems", in IST Transactions of Information Technology- Theory and Applications, Vol. 1, No. 1 (2), ISSN 1913-8822, pp. 1-7, Date Published: July 2010
- Hong Zhang, Rajiv Kishore, Ram Ramesh, "Semantics of the MibML Conceptual Modelling Grammar: An Ontological Analysis Using the Bunge-Wand-Weber Framework", Journal of Database Management, Vol. 18, Issue 1, edited by K. Siau, pp. 1-19, 2007.

- Ahmed. Z D. Gerhard: "How does Ontology Contribute in Semantic Web Development?", In the proceedings of 6th CIIT Workshop on Research in Computing, (CWRC Fall 2007), P 3, 27 October 2007
- Karin Koogan Breitman, Marco Antonio Casanova and Walter Truszkowski, "Semantic Web: Concepts, Technologies and Applications", ISBN 978-1-84628-581-3 (print), Springer-Verlag London Limited
- M.Y. Huang, Y.J. Lin, Hu Xu, "A framework for web-based product data management using J2EE", In Springer-Verlag London Limited,Received: 8 October 2002 / Accepted: 13 March 2003 / Published online: 27 October 2004
- Ahmed. Z, D. Gerhard, " Contributions of PDM Systems in Organizational Technical Data Management ", In the proceedings of The First International Conference On Computer, Control & Communication, Karachi Pakistan, 12-13 November 2007,
- Ahmed, Z. 2009. Intelligent semantic oriented agent based search (I-SOAS). In Proceedings of the 6th international Conference on Frontiers of information Technology (Abbottabad, Pakistan, December 16 - 18, 2009). FIT '09. ACM, New York, NY, 1-4. DOI= http://doi.acm.org/10.1145/1838002.1838065
- T.Ina., Ahmed. Z. 2010. Integration of NLP Search in I-SOAS; Construction of Natural Language Grammar and Mapping with Ontologies.ISBN: 978-3-8433-5851-4, LAP Lambert Academic Publishing Germany, Pages 68.
- Ahmed. Z, Majeed. S, Dandekar.T,. 2010. Towards Design and Implementation of a Language Technology based Information Processor for PDM Systems, IST Transactions of Information Technology- Theory and Applications, Vol. 1, No. 1 (2)ISSN 1913-8822, pp. 1-7, Publisher: IST Press.
- Ahmed.Z. 2010. Proposing LT based Search in PDM Systems for Better Information Retrieval. International Journal of Computer Science & Emerging Technologies (E-ISSN: 2044-6004), Volume 1, Issue 4, P86-100, December 2010

- Ahmed. Z. 2011. Designing Flexible GUI to Increase the Acceptance Rate of Product Data Management Systems in Industry. International Journal of Computer Science & Emerging Technologies, (E-ISSN: 2044-6004), Volume 2. Issue 1., Page 100-109, February, 2011
- C. S. Sung, Sam Joon Park, "A component-based product data management system", In Springer-Verlag London Limited 2006, Received: 18 April 2005 / Accepted: 25 October 2005/ / Published online: 2006
- Miller, E., "What's PMD ?" Computer- Aided Enineering Magzine, September 1997.
- Joseph Galasso, "Analyzing English Grammar: An Introduction to Feature Theory - A Companion Handbook", California State University, Northridge, Draft May 5 2002
- Bruce W. Ballard and John C. Lusth, "An English-language processing system that "learns" about new domains", Pages 39-46, Year of Publication: 1983, ISBN ~ ISSN: 0095-6880, 0-88283-039-2, ACM New York, NY, USA
- ANTLRWorks: The ANTLR GUI Development Environment- Screen shots and movies, last reviewed <http://www.antlr.org/works/screenshots/editor.jpg>
- Leo Sauermann, Gunnar Grimnes, Thomas Roth-Berghofer., "The Semantic Desktop as a foundation for PIM research", Personal Information Management (PIM 2008), CHI 2008 Workshop, April 5-6, 2008, Florence, Italy
- Sam Joseph, "P2P MetaData Search Layers", In Second International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2003), International Conference on Autonomous Agents and MultiAgent Systems, Melbourne Australia, July 14, 2003
- Ahmed Z. "I-HMI and LT-SOS in PDM Systems; Implementing AI Concepts for The Advancement of Multiple Role Based User Oriented Product Data Management Systems", Book Publication, ISBN-13: 978-3-639-31070-2, ISBN-10: 3639310705, EAN: 9783639310702, Language: English, Publishers: VDM Verlag Dr. Muller, Pages: 240, Year Publication 2011.
- Ahmed Z. "Importance of I-SOAS In PDM Community", International Journal of Web Applications, Vol. 3, No.1, Page: 12-16, ISSN: 0974-7710 (Print), 0974-7729 (Online), March 2011

- Ahmed Z. "Designing Knowledge Base towards PDMS", International Journal of Information Technology and Engineering, Vol 2. No. 1, Pages 9-12, International Science Press, ISSN: 2229-7367 (Print only), June 2011
- Sean B. Palmer, "The Semantic Web: An Introduction", viewed February, 28. 2007, <http://infomesh.net/2001/swintro>
- Wernher Behrendt, "Ambient Intelligence Semantic Web or Web 2.0", In Proceedings of Semantic Content Engineering 2005, Salzburg 2005
- Sebastian Ryszard Kruck, Mariusz Cygan, Piotr Piotrowski, Krystian Samp, Adam Westerski and Stefan Decker, "Building a Heterogeneous Network of Digital Libraries on Semantic Web", In Proceedings of Semantic Systems From Visions to Applications 2006, Vienna Austria 2006
- B. Omelayenko., "Learning of ontologies for the Web: the analysis of existent approaches", In Proceedings of the International Workshop on Web Dynamics, held in conj. with the 8th International Conference on Database Theory (ICDT'01), London, UK, 3 January 2001
- Gerhard Weikum, Jens Graupmann, Ralf Schenkel, and Martin Theobald, Towards a Statistically Semantic Web, P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 3–17, 2004. Springer-Verlag Berlin Heidelberg 2004
- Grit Denker, Son Nguyen, and Andrew Ton, "OWL-S Semantics of Security Web Services: a Case Study", J. Davies et al. (Eds.): ESWS 2004, LNCS 3053, pp. 240–253, 2004. sSpringer-Verlag Berlin Heidelberg 2004
- Witold Abramowicz, Tomasz Kaczmarek, and Krzysztof W˛ecel, "How Much Intelligence in the SemanticWeb?", P.S. Szczepaniak et al. (Eds.): AWIC 2005, LNAI 3528, pp. 1–6, 2005,. Springer-Verlag Berlin Heidelberg 2005
- D. Fensel. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Berlin, 2000
- Okkyung Choi, SeokHyun Yoon, Myeongeun Oh, and Sangyong Han, "Semantic Web Search Model for Information Retrieval of the Semantic Data", C.-W. Chung et al. (Eds.): HSI 2003, LNCS 2713, pp. 588-593, 2003., Springer-Verlag Berlin Heidelberg 2003

- Jos´e Saias and Paulo Quaresma, "A Methodology to Create Legal Ontologies in a Logic Programming Information Retrieval System", V.R. Benjamins et al. (Eds.): Law and the Semantic Web, 3369, pp. 185–200, 2005. Springer-Verlag Berlin Heidelberg 2005
- Amalia Todirascu, Laurent Romary, and Dalila Bekhouche, "Vulcain – An Ontology-Based Information Extraction System" B. Andersson et al. (Eds.): NLDB 2002, LNCS 2553, pp. 64–75, 2002. Springer-Verlag Berlin Heidelberg 2002
- Marc Ehrig and York Sure, "Ontology Mapping - An Integrated Approach", J. Davies et al. (Eds.): ESWS 2004, LNCS 3053, pp. 76–91, 2004. Springer-Verlag Berlin Heidelberg 2004
- Ahmed. Z, Ali. M and Saman. M, "Implementing Computerized and Digitally Mobile Home Automation System towards Electric Appliance Control and Security System", International Journal of Emerging Sciences, ISSN 2222-4254, Vol. 1, No. 3, Pages 487-503, Paper ID 10, Category Regular Papers, September 2011.
- Ahmed. Z, Saman. M, "Machine Learning and Data Optimization using BPNN and GA in DOC", International Journal of Emerging Sciences, Vol. 1, No. 2, Pages 108-119, Paper ID 6, Category Regular Papers, ISSN: 2222-4254, June 2011.
- Saman. M, "Towards the Contribution of NEMDBs in Global Genetic Heterogeneity", International Journal of Emerging Sciences (IJES), Vol 1, No. 3, Pages 433-443, ISSN: 2222-4254, Paper ID 6, Category Regular Papers, September 2011.
- Ahmed. Z, Saman. M, "Towards Increase in Quality by Preprocessed Source Code and Measurement Analysis of Software Applications", International Science & Technology Transactions on Information Technology- Theory and Applications, Vol.1,No. 1, pp.8-13, ISSN 1913-8822, September 2010.
- Ahmed. Z, "Towards Performance Measurement and Metrics based Analysis of PLA Applications", International Journal of Software Engineering & Applications, Vol.1, No. 3,pp 66-80, ISSN: 0975 - 9018 (Online), 976-2221 (Print), July 2010.
- Ahmed. Z, Saman. M, "NLP Syntax Structure using ANTLR in I-SOAS", International Journal of Information Technology and Engineering, Vol 2. No. 2, ISSN: 2229-7367 (Print only) 2011.

- Ahmed. Z, Tacheva. T, , "Integration of Natural Language Processing towards Semantic Oriented Search", International Journal of Computer Science and Software Technology, Vol. 3, No.2, pp. 69-76, International Sciences Press, ISSN: 0974-3898 (Print only), July-December 2010

Authors Bibliography:

Thomas Dandekar is the head of the department of bioinformatics, Biocenter, Am Hubland, University of Wuerzburg Wuerzburg Germany, combining genome-based bioinformatics with systems biological modelling.

Zeeshan Ahmed is the software engineer, researcher at the department of bioinformatics, Biocenter, Am Hubland, University of Wuerzburg Wuerzburg Germany, developing intelligent software systems towards product data analysis, visualization and management.

Saman Majeed is the doctoral scientist the department of bioinformatics, Biocenter, Am Hubland, University of Wuerzburg Wuerzburg Germany.